Monitoring and Modeling Performance of Communications in Computational Grids

NAS Technical Report NAS-03-016
November 2003

Michael A. Frumkin*, Thuy T. Le**
*NASA Advanced Supercomputing (NAS) Division
NASA Ames Research Center, Moffett Field, CA 94035-1000
**San Jose State University
frumkin@nas.nasa.gov, thuytle@email.sjsu.edu

Abstract

For efficient use of a computational grid, which includes machines located in a number of sites, we have to be able to estimate data delivery times between the machines. For dynamic distributed grids it is unrealistic to know exact parameters of the communication hardware and the current communication traffic, hence we should rely on a performance model of the network to estimate data delivery times.

We build an empirical model based on observation of message delivery times with various message sizes and time scales. Our experiments show presence of multiple bands in the histogram of the logarithm of message delivery times. The histograms represent a multiband model reflecting multiple paths and probabilities the messages can travel between the grid machines.

Keywords: grid computing, message, delivery time, measurement, benchmarks.

1 Introduction

The quality of assignment of the application tasks to the grid machines, also known as dynamic scheduling, or navigation, directly affects application turnaround time. The assignment decisions depend on many factors: identification of appropriate grid machines, the load of the machines, the application requirements, the latency and bandwidth of the communication network, and the network traffic. For estimation of these factors a number of tools are available, including `traceroute`, the *Network Weather Service* (NWS), and the *NAS Grid Benchmarks* (NGB) [12, 2]. These tools allow estimation of these factors on a sparse subset of possible grid loads.

We use observations of message delivery times in a computational grid to build an empirical multiband model of the network which generalizes a singleband model of [4]. This model can be used for quick estimation of the time it takes to communicate application data between hosts of a grid. We build the model in two steps. First, we obtain experimental data for the message delivery time between hosts. Then, we extract the band structure of these communications by calculating a histogram of the logarithms of message delivery times.

We use a Java version of the NAS Grid Benchmarks [2, 7] as a measurement tool since it has a number of advantages. The Java version is architecture and OS neutral and can be easily used to build a computational grid environment. It does not require users to have accounts on all grid machines, which simplifies our collaborative effort. Assigning benchmark tasks can be done by making simple changes in a benchmark data flow graph. This provides great flexibility to concentrate on any interesting subset of the hosts. The benchmarks can be executed in a monitoring mode to build a database of the grid measurements including one way message transmission time. Computational grids use many different mechanisms for communication between machines, including `MPI`, `Java RMI`, `GridFTP`, and `scp`. To verify the qualitative results obtained with NGB we compare the results with the measurements obtained with `traceroute` and with `scp`.

2 Observing the Network Traffic

2.1 Tools for Observing Network Traffic

The `traceroute` is the most popular tool for discovering the network structure, for finding latencies incurred by the messages sent between machines, and for diagnostics of the network anomalies [8]. For testing a route between hosts A and B `traceroute` sends a sequence of test packets (UDP datagram) from A to B, until a packet reaches its destination [9], Section 25.6. The first packet has value of IPv4 *Time To Live* (TTL) field (or IPv6 hop limit field) equal to onr. This packet causes the first router along an (A, B) path to return "time exceeded in transit" error. The value of the TTL field of the next packet is incremented by one. Each router along the path of the packet decrements TTL by one, hence each packet travels one hop farther than the previous one. If a packet reaches B, the host returns "port unreachable" error. The returned error messages allow to find out the IP addresses of the routers where TTL vanishes. The `traceroute` prints these IP addresses and the time elapsed since sending a packet till receiving an error message "the packet has not been delivered".

A number of `traceroute` servers have a web page that allows to find the routers and the latencies along a path from the servers to other internet hosts. The `traceroute` also allows to determine the bandwidth of an (A, B) path for the messages which sizes fit $[40, 65534]$ interval. In [8] they use extensive experiments with `traceroute` to detect and classify internet anomalies. In some papers they provide evidence that the arrival times of network messages are fractal, i.e. self-similar in a range of time scales [11] and have heavy tails in the distribution of arrival times. One conclusion from this self-similarity is that the arrival times are bursty in an interval of time scales.

Information on TCP/IP performance (latency and bandwidth) can be obtained with the Network Weather Service (NWS). The NWS monitors delivery times of the messages sent between participating network hosts. The measurements obtained are then used by the NWS to estimate the latency and bandwidth and to make predictions of these characteristics. The NWS does not compensate for the clock skew between hosts. It makes the observation of the one directional message delivery time unreliable and prevents the measurement of network asymmetry.

A number of tools were developed to extend the ability of `traceroute` to display information about the network. These tools, including *3D Traceroute*, [10] and *GridMapper* [1] are able to collect statistics about network latency and bandwidth, to obtain the geographical information about the hosts and routers and visualize the statistics and the network activity. The GridMapper can access performance information sources and map domain names to physical locations. It enables visualization of layout of the grid and animation of activity of the grid hosts and networks.

A direct copy of a file across the network by using `ftp` (or `scp` for secure networks) can be used to collect statistics on time to to copy files between grid hosts. If called from the receiving machine the `scp` is blocking, the execution time of `scp` can be used for measuring time to copy files.

2.2 Complexity Levels of Network Traffic

The root of the difficulties in understanding the network traffic is the numerous sources of uncertainty and even pathology in the networks. There are four main categories of uncertainty affecting network traffic: topology, metric, events, and timing. The topological uncertainty affects the (A, B) path taken by different packages of the same message. The path can have fast fluctuation (faltering), loops, and temporary outages (loss of network connectivity) [8]. The metric uncertainty affects packet delivery time due to varying load on the hosts, routers, and other network hardware (switches, exchange points). The events uncertainty affects the sharing of the network elements by different messages. Since many events causing internet traffic are external to the network (do not have a causal relation to any network event), a message from A to B will have unknown interference with other messages along its way. And, finally, message timing, i.e., a small variation in the timing between different events can substantially affect the message delivery time. For illustration we show two aspects of the metric uncertainty: network asymmetry and violation of the triangle inequality.

The methods of traffic routing do not claim that either the time or the path which a packet travels from A to B are the same as those of a packet that travels from B to A. Moreover, one can observe that (A, B) and (B, A) routes can be different. For example, at the time this paper was written (May 2003), the route from `www.slac.stanford.edu` (A) to `www.above.net` (B) has 9 hops, while the route n the opposite direction has 13 hops. Surprisingly, the roundtrip times measured from each site are very close to each other, indicating that a packet sent from A to B is returning back to A along a different route.

The unpredictability of the round-trip times becomes evident when we look at the output of a single call to the `traceroute`. Quite often, the packets with larger TTL return before the packets with smaller TTL.

Table 1 which contains `traceroute` measurements between three websites demonstrates that:

$$time(ABOVE.NET, SLAC) + time(SLAC, NAS) <$$
$$< time(ABOVE.NET, NAS)$$

which is a violation of the triangle inequality. The routing tables are supposed to be built with use of the shortest path tree from each router to each network. In an ideal situation, this would guarantee the triangle inequality. In reality, the triangle inequality is violated due to use of a different "length" function for building the shortest path trees on different routers.

3 Using the NAS Grid Benchmarks for Monitoring the Grid

3.1 The NAS Grid Benchmarks

The *NAS Grid Benchmarks* (NGB) [2] were designed to test grid functionality, performance of grid services, and to represent typical grid applications. NGB use a basic set of grid services such as create task and communicate. An NGB instance is specified by a *Data Flow Graph* encapsulating NGB tasks (NAS Parallel Benchmark (NPB) codes) and communications between these tasks. Currently, there

Table 1. Roundtrip times between `www.above.net`, `www.slac.stanford.edu`, and `www.nas.nasa.gov`. The table entries show the average time while the numbers in parentheses show the maximum deviation (three measurements per table entry on two consecutive days).

Server Name	SLAC	ABOVE.NET	NAS
SLAC	-	55.9 (0.4)	2.4 (0.1)
ABOVE.NET	55.8 (0.3)	-	68.7 (1.3)
NAS	3.5 (1.2)	68.1 (1.2)	-

are four types of NGB: Embarrassingly Distributed (ED), Helical Chain (HC), Visualization Pipeline (VP), and Mixed Bag (MB). For this study we use a Java version of the HC.S benchmark, Figure 1.

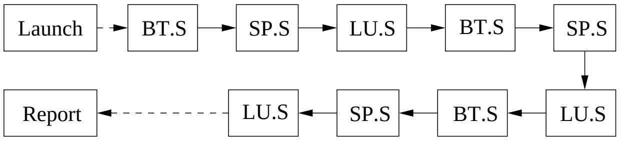

Figure 1. The data flow graph of HC, class S benchmark. Solid and dashed arrows signify data and control flow respectively.

3.2 Enhancements to the NGB

For probing and analysis of the message delivery times some enhancements to the NGB were necessary. We added monitoring capabilities that enabled us to run the benchmarks periodically and save the monitoring results in a database. We improved granularity of NGB message sizes and added a clock synchronization.

The amount of data the benchmarks tasks send to their successors varies from 69KB (Class S) to 245MB (class C) which is too sparse for observing delivery times. Our first modification was to add some extra data to the array sent by a task to its successors. The message sizes were made controllable by an input parameter to the utility `ngbrun` used to submit the benchmarks. This flexibility of choosing the message size made it possible to implement grid monitoring with growing message sizes, Section 4.

Synchronizing of clocks of the computers in computational grids is accomplished by means of the (Simplified) Network Time Protocol ((S)NTP). The SNTP allows synchronization of the computers in a WAN within accuracy of tens of milliseconds [6]. However, increasing speed of the network routers and switches allows fast messages between grid machines, so even a 10ms clock skew becomes noticeable. Another source of the time skew is improper function/configuration of NTP daemons. In our experiments the clock skew between grid machines manifested itself by the negative communication time.

We implemented a clock synchronization mechanism which has accuracy equal to half of the roundtrip time of the time stamps and on average, reduces the clock skew to less than 20ms. The clock synchronization task works in asymmetrical networks (networks where the time to send a message from A to B may differ from the time to send the same message from B to A). Correction of the clock skew enables measurement of asymmetry of the networks in which message delivery times are significantly bigger than half of roundtrip time.

We describe a synchronization of a pair of machines A and B. To synchronize the whole grid, we build a spanning tree of the network, then choose a root of the tree and apply the pairwise synchronization to each edge of the tree starting from the root. We use a synchronization mechanism employed by NTP [6] by sending a time stamp $t_A(1)$ from A to B and a time stamp $t_B(2)$ from B to A and record their arrival times $t_B(1)$ and $t_A(2)$ respectively. Let c_{AB} and c_{BA} be the time stamp delivery times (measured by some external clock which we use only for justification of our synchronization mechanism). Let $c = c_{AB} + c_{BA}$ and $C = t_B(1) - t_A(1) + t_A(2) - t_B(2)$ be round-trip and observed round-trip of a time stamp respectively. Also, let $\delta = c_{AB} - c_{BA}$ and $\Delta = t_B(1) - t_A(1) - (t_A(2) - t_B(2))$ be the asymmetry and observed asymmetry of the delivery time. Let w be the time the clock in B running ahead of clock in A, which we assume to be a constant during the period of time c. We have the following relations:

$$t_B(1) - t_A(1) = c_{AB} + w, t_A(2) - t_B(2) = c_{BA} - w$$

hence, $c = C$ and $2w = \Delta - \delta$. We add $\Delta/4$ to the clock in A, and subtract it from the clock in B. If $\delta = 0$ this correction will synchronize the clocks in A and B. In any case the accuracy of the synchronization will be $\delta/2 \leq c/2 = C/2$.

4 Experimental Results

For our experiments we used a grid with hosts located at NASA Ames and NASA Glenn research centers, Figure 2. The Java version of the NGB with the enhancements was installed on these grid machines. It uses the Java Registry to register and lookup task services on the hosts, and it uses and the Java Remote Method Invocation (RMI) to access the services, run benchmark tasks, and communicate data between tasks. In addition to the HS.S we used the `traceroute` and the `scp`. The typical time to execute the HC.S benchmark varied between 20 and 40 seconds in our setup.

We monitored the grid over periods of 24-48 hours for a few weeks in April-May 2003. We used two types of monitoring: with fixed and growing message sizes, Table 2. The interval between successive runs ranged from 1 to 30 minutes. Since we have not observed an essential difference over this range, we show monitoring results with 10 min-

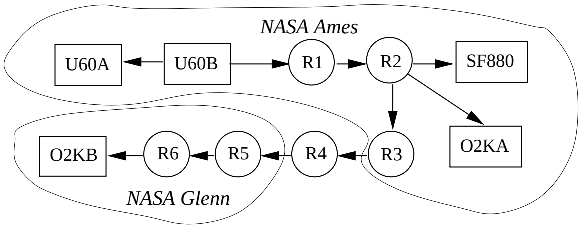

Figure 2. The experimental grid and its spanning tree. U60A and U60B are 2 processor, 450MHz ULTRA60 from SUN. O2KA and O2KB are 32 and 24 processor, 250MHz Origin2000 machines from SGI. SF880 is 8 processor, 900 MHz UltraSparc 3 from SUN. The routers R1 through R6 were identified by the `traceroute` ran from U60B.

utes intervals, unless otherwise specified.

Table 2. The message sizes used to monitor message delivery time on the grid.

Monitoring Tool	initial size	increment	final size
NGB	69 KB	0	69 KB
NGB	69 KB	400 KB	40 MB
`traceroute`	40B	0	40 B
`traceroute`	64 KB	0	64 KB
`scp`	1 MB	1 MB	40 MB

To build a model of delivery times for large messages, we performed monitoring of the grid with growing message sizes (bandwidth experiments). We started with a series of experiments between U60A and O2KA. The results of experiments using `scp` and HC.S are shown in Figure 3. The histograms of the logarithm of the message size over delivery time (i.e. of the logarithm of observed bandwidth) have multiple extremal points and well-defined bands (also shown in Figure 3).

Typical monitoring results with fixed message size and the histograms of logarithm of delivery times are shown in Figure 4. All the histograms have multiple extremal points and well-pronounced band structures. The results of the bandwidth experiments involving all five grid machines are shown in Figure 5. A number of the plots have well separated bands. Notice asymmetry in the grid, for example, the delivery time O2KA ⇒ SF880 is twice as long as the delivery time O2KA ⇐ SF880 .

5 Analysis of the Experimental Results

A statistical analysis of the message delivery time obtained with NGB, `traceroute`, and `scp` shows that, typically, histograms of the logarithm of message delivery times have well separated peaks, as there are multiple paths between the hosts, and each message travels along one of these paths. If the shortest path is available, the message passes it, otherwise, if the second shortest path is available, the message passes it and so on. On the other hand, the `traceroute` shows that the routes between hosts in our grid remain stable.

To explain the presence of the peaks, we consider the access time of a word located in memory of R10000 processor. There are five possible cases depending on the location of the word [3]. The word can be in the registers, L1 cache, L2 cache, main memory where the page address is in TLB, and main memory where the page address is not in TLB. Depending on the case, the access time will be 0, 2-3, 8-10, 75-250, or 2000 machine cycles. The histogram of the access time would have five peaks each having a small spread relative to the separation of peaks. The logarithm of the delivery time compensates for scaling of the access times.

A delivery channel for a message from host to host involves a sequence of devices each having its own scale and bands. Depending on the state of a device, a message passes through one of its bands. So the length of the base band of the channel is a sum of the base bands of the devices. The second band is a sum of the base bands of all devices except one with the slowest second band and so on. The separation between bands becomes smaller as the lower bands become occupied.

6 Conclusions

We have measured delivery times over 4K messages between hosts of a two-site computational grid using three different tools to originate and monitor the messages. The experiments have shown consistency over a time interval of eight weeks. The difference in the message delivery time obtained with `traceroute` and NGB is due to the overhead of interpretation of Java used to implement NGB and to the fact that the `traceroute` does not count the delay in the response time of the sending side.

The statistical behavior of the results obtained by all three methods show splitting of the histogram of the logarithm of message delivery time into multiple bands. The presence of the bands we explain by the different modes the messages travel form host-to-host. The base band is observed in the case where there is no interference with network traffic or host processes. As the traffic gets thicker and the host load increases, the base band becomes occupied and the messages are forced through a slower band. We believe that this model of message delivery explains the presence of heavy tails in the distribution of the message delivery times [5] and answers a question raised in [11], Section 4 regarding the origin of the heavy tails.

References

[1] W. Allcock, J. Bester, J. Bresnahan, I. Foster, J. Gawor, J.A. Insley, J.M. Link, M.E. Papka. *GridMap-*

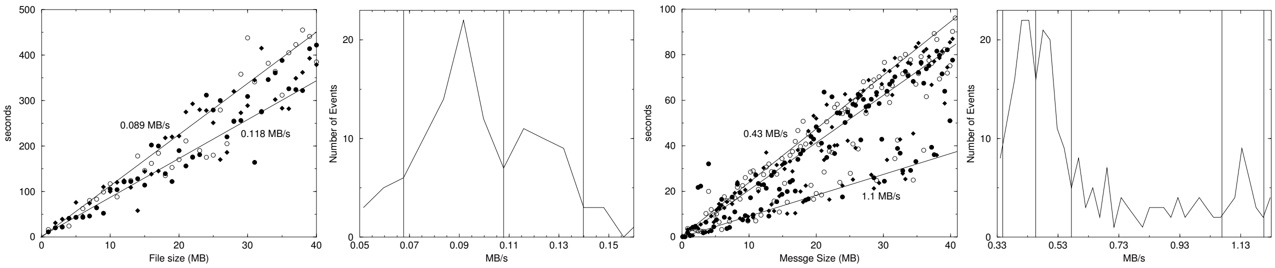

Figure 3. Three sets of experiments on time to copy files U60A ⇒ O2KA using `scp` (left two plots), and to deliver a message using HC.S (right two plots). In the first case the file sizes were in the range [1,40] MB with 1 MB increments, and the histogram shows 2 bands of 0.089 MB/s and 0.118 MB/s. In the second case the message sizes were in the range [0.069,40] MB with 0.4MB increments, and the histogram shows 3 bands of 0.43 MB/s. 0.49 MB/s, and 1.1 MB/s.

per: A Tool for Visualizing the Behavior of Large-Scale Distributed Systems. Proceedings of HPDC11, 23-26 July 2002, Edinburgh, Scotland, pp. 179-187.

[2] M. Frumkin, Rob F. Van der Wijngaart. *NAS Grid Benchmarks: A Tool for Grid Space Exploration.* Cluster Computing, Vol. 5, pp. 247-255, 2002.

[3] M. Frumkin, H. Jin, J. Yan. *Automation of Data Traffic Control on DSM Architectures.* LNCS 2074, p. 771-780.

[4] Thuy T. Le. *Evaluating Communication Performance Measurement Methods for Distributed Systems*. Proceedings of the 14th IASTED International Conference “Parallel and Distributed Computing and Systems” (PDCS’2002), Cambridge, USA, Nov 4-6, 2002, pp. 45-51.

[5] W.E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson. *On the Self-Similar Nature of Ethernet Traffic.* IEEE/ACM Transactions on Networking 2(1994), pp. 1-15.

[6] D.L. Mills. *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI.* RFC 2030 (http://rfc.net), 18 pp., 1996.

[7] The NAS Grid Benchmarks. http://www.nas.nasa.gov.

[8] V. Paxson. *End-to-End Routing Behavior in the Internet.* IEEE/ACM Transactions on Networking 5(5), pp. 601-615, 1998.

[9] W.R. Stevens. *UNIX Network Programming.* Vol. 1, Prentice-Hall International, Inc. 1998.

[10] *3D Traceroute.* www.hlembke.de/prod/3dtraceroute/.

[11] W. Willinger, V. Paxson, M.S. Taqqu. *Self-Similarity and Heavy Tails: Structural Modeling of Network Traffic.* A Practical Guide to Heavy Tails: Statistical Techniques and Applications. R. Alder, R. Feldman, M. S. Taqqu, Ed. Birkhauser, Boston 1998.

[12] R. Wolski, N.T. Spring, J. Hayes. *The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing.* http://nws.npaci.edu/NWS/.

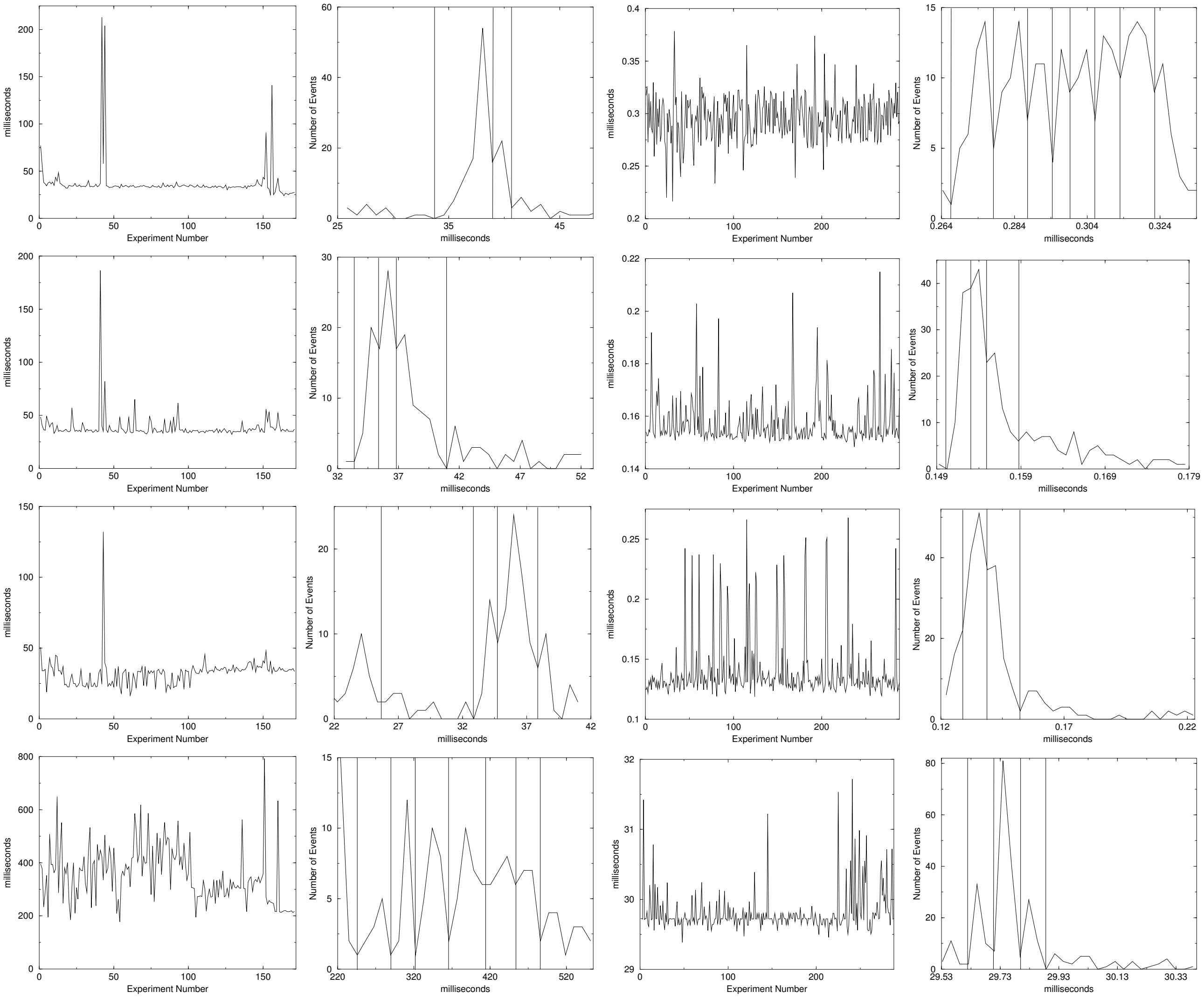

Figure 4. The first column shows message delivery times U60A ⇒ O2KA , U60B ⇒ O2KA , O2KA ⇒ SF880 , and SF880 ⇒ O2KB obtained by running HC.S of the NAS Grid Benchmarks with 69KB messages. The histograms in the second column show the bands in the distribution of the logarithm of message delivery times (back scaled to the time). The third column shows half of roundtrip time of 40 B messages between machines U60A ⇔ O2KA , U60A ⇔ SF880 , U60A ⇔ U60B , and U60A ⇔ O2KB obtained by `traceroute`. The histograms in the fourth column show bands in the distribution of the logarithm of message roundtrip times (back scaled to the time).

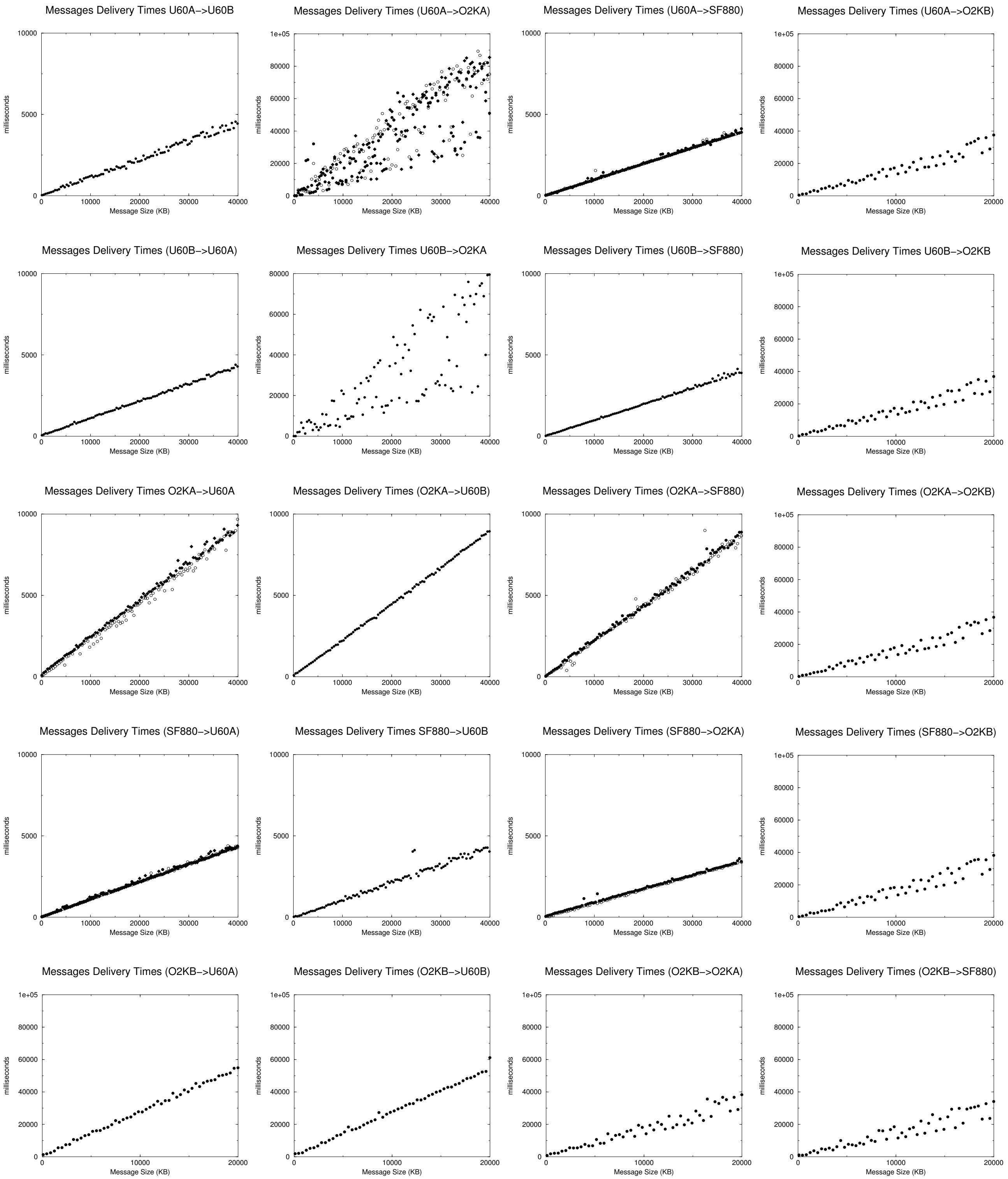

Figure 5. Message delivery times between U60A , U60B , O2KA , SF880 , and O2KB obtained with the HC.S and the messages varying 69KB-40MB in size (note 10x difference in Y-scale on the plot of times from U60A , U60B to O2KA). On some plots, symbols indicate different experiments.